AI Programming A3
Report: Configurable Machine Learning Classifier Framework

This project presents a configurable machine learning (ML) framework designed to support multiple classifiers and datasets. Currently, it includes implementations for Convolutional Neural Networks (CNNs) with ResNet-14, Logistic Regression, and Support Vector Machines (SVMs). The framework supports two datasets—CIFAR-10 and an obesity dataset—and is designed for extensibility, enabling easy integration of new classifiers and datasets.

```python
def main():
    parser = argparse.ArgumentParser(description="Select classifier type")

    # Define the argument for classifier
    parser.add_argument(
        'classifier',  # Command-line argument name
        choices=[e.name for e in ClassifierType],  # Enum names as valid choices
        help="Classifier type to use. Options: CNN, SVM, LogisticRegression"
    )

    parser.add_argument(
        'data_type',  # Command-line argument name
        choices=[e.name for e in DataType],  # Enum names as valid choices
        help="Data type to use. Options: TABULAR, IMAGES, TEXT, TEXT_IMAGES"
    )

    parser.add_argument(
        'dataset_name',
        help="Data set to be used Options: cifar, obesity"
    )

    parser.add_argument(
        'data_path',
        help="Data set to be used it should be directory path for images and csv path for the tabular data"
    )
```

Framework Design

The architecture centers on two main abstractions:

1. **Data Preprocessor**: Handles data preprocessing for different dataset types.
2. **Object Classifier**: Defines the structure and behavior of ML classifiers.

1. Data Preprocessor

The `DataPreprocessor` class is an abstract base class (ABC) that ensures consistent preprocessing across datasets. Each dataset implementation must define:

- **`preprocess_data`**: Prepares the entire dataset for training and evaluation.
- **`preprocess_sample`**: Prepares individual samples for prediction.

```
class DataType(Enum):
    TABULAR = 1
    IMAGES = 2
    TEXT = 3
    TEXT_IMAGES = 4


class DataPreprocessor:
    def __init__(self, dataset_name: str, data_path: str):
        self.dataset_name = dataset_name
        self.data_path = data_path
    @abstractmethod
    def preprocess_data(self):
        pass
    @abstractmethod
    def preprocess_sample(self, sample):
        pass
```

2. Object Classifier

The `ObjectClassifier` class is another ABC that ensures all classifiers adhere to a consistent structure. The following methods must be implemented:

- **`preprocess_data`**: Invokes the data preprocessor to prepare the data.
- **`build_model`**: Defines the model architecture.
- **`train_model`**: Handles the training process.
- **`evaluate_model`**: Assesses the model's performance on test data.
- **`predict`**: Makes predictions on new data samples.

```
class ClassifierType(Enum):
    CNN = 1
    SVM = 2
    LogisticRegression = 3


class ObjectClassifier:

    def __init__(self, data_preprocessor: DataPreprocessor):
        self.data_preprocessor = data_preprocessor
    @abstractmethod
    def preprocess_data(self):
        pass
    @abstractmethod
    def build_model(self):
        pass
    @abstractmethod
    def train_model(self):
        pass
    @abstractmethod
    def evaluate_model(self):
        pass
    @abstractmethod
    def predict(self, samples: list) -> list:
        pass
```

Implementation Details

1. CIFAR-10 Dataset (Image Classification)

Data Type: Images
Model: ResNet-14 (CNN)

Data Preprocessing

The CIFAR-10 dataset was preprocessed with the following transformations:

- Normalization of pixel values.
- Random cropping and flipping for data augmentation.
- Conversion to tensors for PyTorch compatibility.

Training and Evaluation

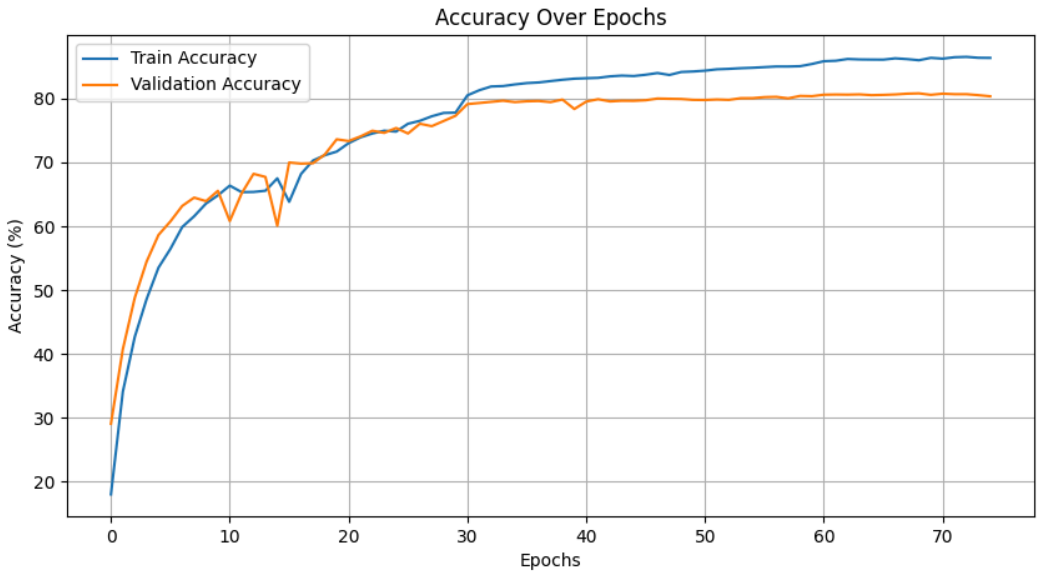
ResNet-14, a lightweight version of ResNet, was trained on CIFAR-10 using the following parameters:

- **Optimizer**: SGD with momentum.
- **Learning Rate Scheduler**: Cosine Annealing.
- **Loss Function**: Cross-Entropy Loss.

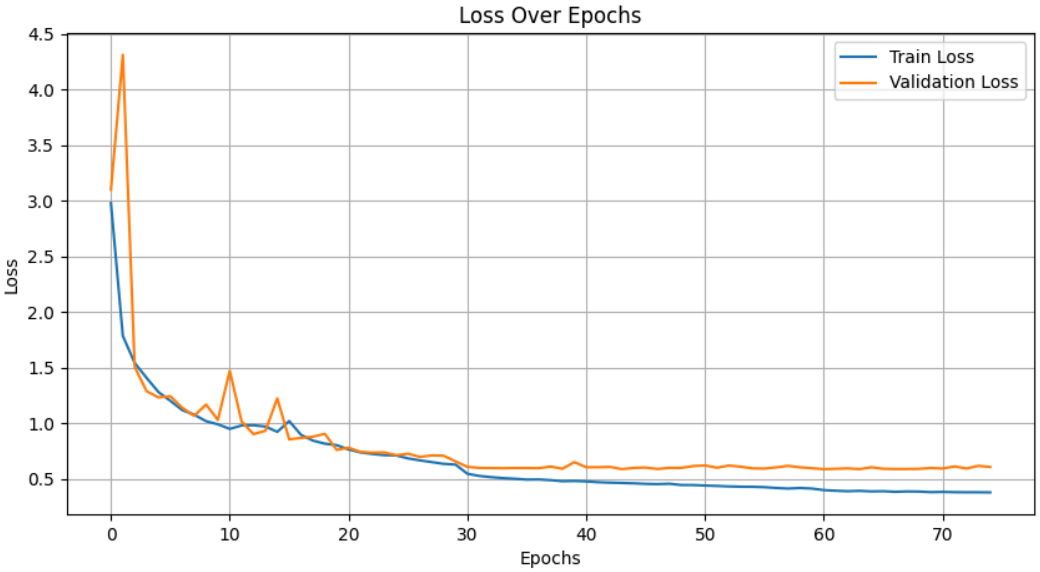
Performance Metrics

The model achieved high accuracy, and the following graphs illustrate its performance over epochs:

- **Accuracy Over Time**:



- **Loss Over Time**:



2. Obesity Dataset (Tabular Data Classification)

Data Type: Tabular
Models: Logistic Regression and SVM

Data Preprocessing

The obesity dataset, provided in CSV format, was preprocessed by:

- Handling missing values.
- Scaling numerical features to a standard range (e.g., 0 to 1).
- Encoding categorical variables using one-hot encoding.

Training and Evaluation

- **Logistic Regression**: A linear model optimized using Stochastic Gradient Descent.
- **SVM**: A kernel-based model trained using a radial basis function (RBF) kernel for non-linear separability.

Performance Metrics

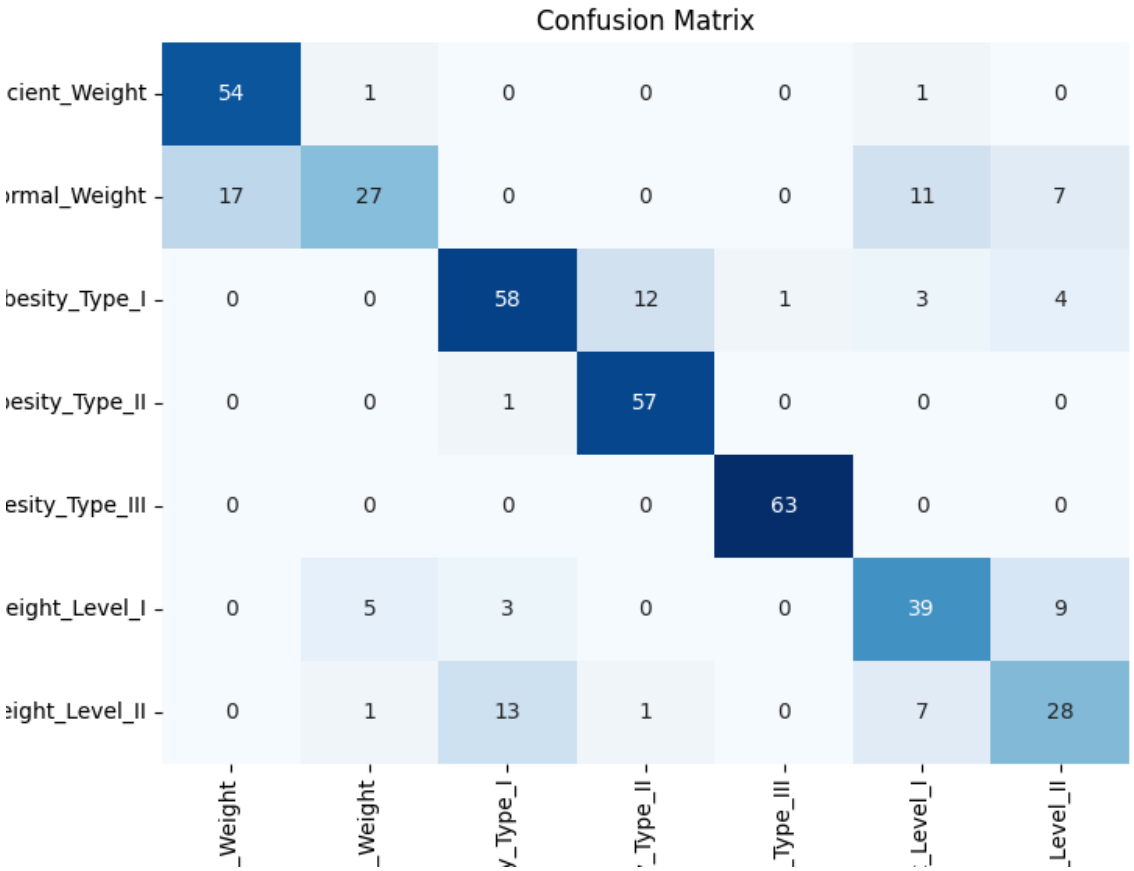
Both models were evaluated using metrics like accuracy, precision, and recall. Here are the comparative results:

- **Logistic Regression Results**:

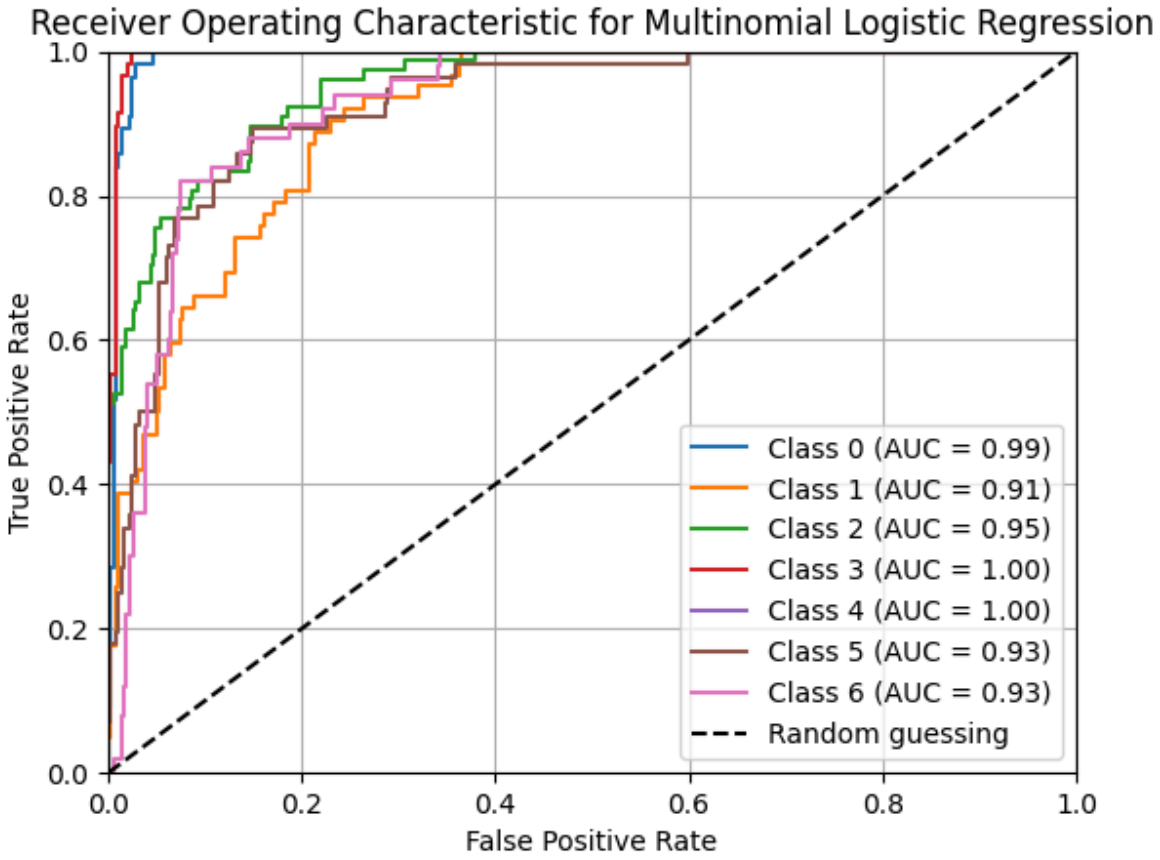
  Accuracy: 0.77

  F1 Score: 0,76

  Confusion Matrix:



ROC Curve

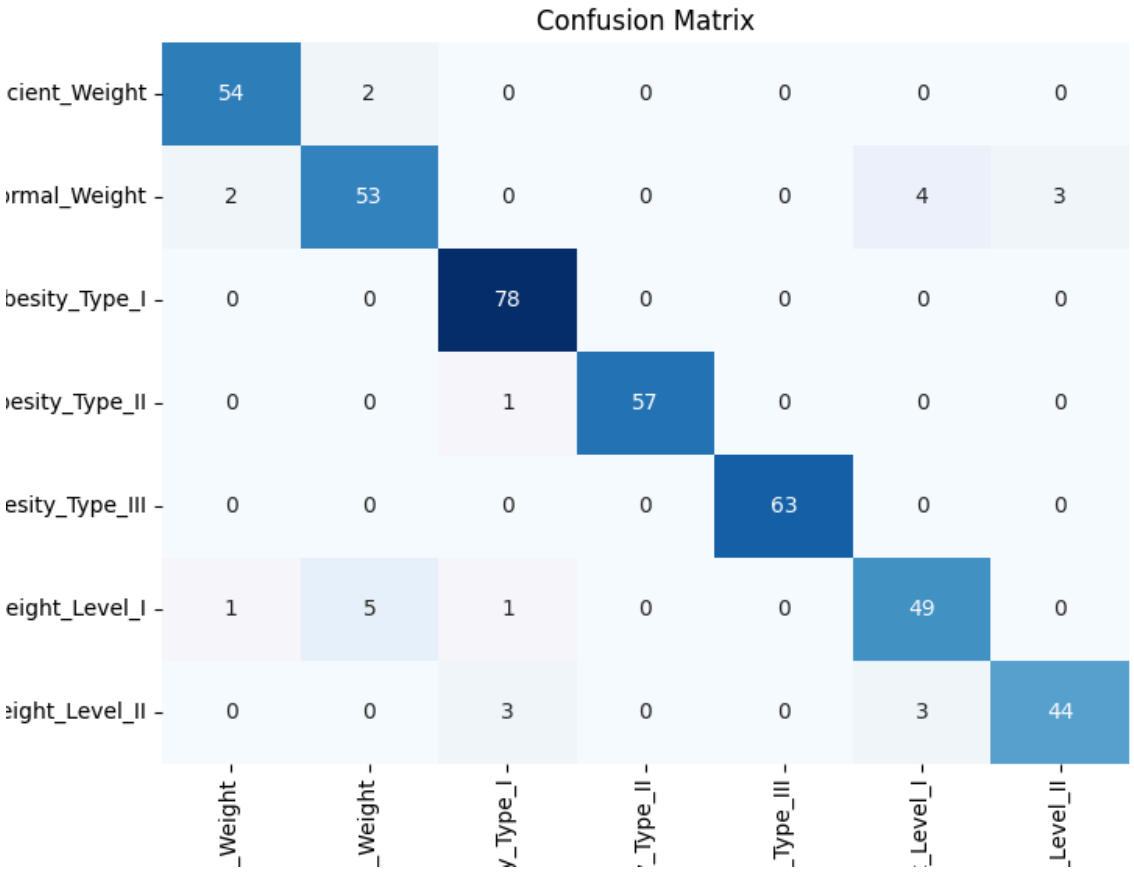

Receiver Operating Characteristic for Multinomial Logistic Regression
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
Class 0 (AUC = 0.99)
Class 1 (AUC = 0.91)
Class 2 (AUC = 0.95)
Class 3 (AUC = 1.00)
Class 4 (AUC = 1.00)
Class 5 (AUC = 0.93)
Class 6 (AUC = 0.93)
Random guessing

- **SVM Results**:

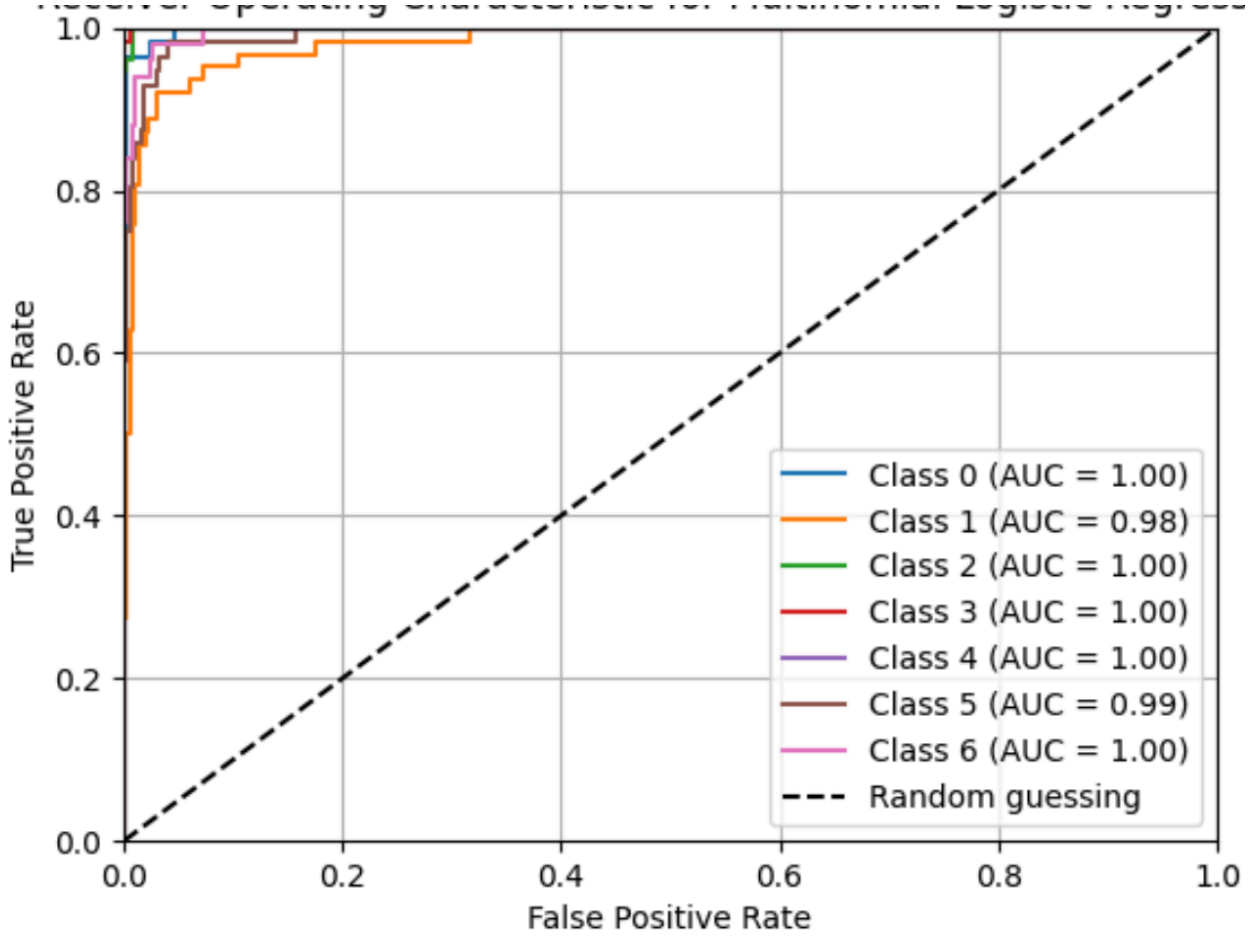
Accuracy: 0.94

F1 Score: 0.94

Confusion Matrix:



ROC Curve


True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
Class 0 (AUC = 1.00)
Class 1 (AUC = 0.98)
Class 2 (AUC = 1.00)
Class 3 (AUC = 1.00)
Class 4 (AUC = 1.00)
Class 5 (AUC = 0.99)
Class 6 (AUC = 1.00)
Random guessing

Extensibility

The framework is designed to allow the addition of:

1. **New Datasets**: Implement the `DataPreprocessor` class for the new dataset.
2. **New Classifiers**: Implement the `ObjectClassifier` class for the new model.

Conclusion

This framework provides a modular and extensible solution for training machine learning classifiers on diverse datasets. It supports various ML models and data types, ensuring flexibility and scalability for future extensions. Current results on CIFAR-10 and the obesity dataset demonstrate the effectiveness of this architecture, achieving competitive performance metrics with both CNNs and classical ML models.: